Tecnologie Web T
Simulazione Compito

Tempo a disposizione: 3 ore

La soluzione comprende la consegna elettronica dei seguenti file mediante l'apposito applicativo Web **esamix** (**http://esamix.labx**):

Biblioteca.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Casa.zip	file zip contenente il sorgente java/class e pagine Web per punto 2
Libro.zip	file zip contenente il sorgente java/class e file XML per punto 3

Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto

N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), equamente distribuiti sui tre esercizi, ovvero almeno 6 punti sul primo esercizio, 6 punti sul secondo esercizio e 6 punti sul terzo esercizio

---

Studenti in debito di Tecnologie Web L-A

Viene richiesto lo svolgimento dei soli esercizi 1 (17 punti) e 2 (16 punti). Tempo a disposizione: 2 ore.
I 18 punti necessari per l'ammissione all'orale sono così distribuiti: almeno 10 punti sul primo esercizio e almeno 8 punti sul secondo

---

ESERCIZIO 1 (11 punti)

Si realizzi una applicazione Web per la gestione del catalogo di una biblioteca basandosi principalmente sulle tecnologie Java server-side Servlet e JSP. L'obiettivo è quello di gestire l'inserimento di un nuovo libro in catalogo (si trascuri la gestione del numero di copie a disposizione) e di cercare i libri in catalogo relativi ad uno specifico autore.

La pagina **nuovoLibro.jsp** permette di inserire un nuovo libro in catalogo. L'utente deve poter inserire il nome del libro, l'autore ed il codice ISBN. Inoltre l'utente deve avere la possibilità di salvare temporaneamente i dati inseriti senza che questi vengano inseriti nel catalogo. In particolare, un pulsante **save** permette di salvare quanto digitato fino a quel momento (ad esempio solo il nome dell'autore ma non il titolo ed il codice ISBN); se l'utente passa alla pagina **elencoLibri.jsp** e poi ritorna alla pagina **nuovoLibro.jsp**, i dati inseriti precedentemente e memorizzati tramite pulsante **save** devono essere inseriti automaticamente nel form della pagina JSP. Se invece l'utente chiude il browser, tali dati vanno persi.
Il libro viene aggiunto nell'elenco solamente se l'utente preme il pulsante **finalizza**; a questo punto il libro è effettivamente disponibile in catalogo da parte di tutti gli utenti del sistema. Dopo aver premuto il pulsante **finalizza** l'utente può inserire un nuovo libro.
Le richieste devono essere gestite da una apposita Servlet **aggiungiLibro**.

La pagina **elencoLibri.jsp** permette di trovare l'elenco dei libri scritti da un particolare autore specificato tramite un opportuno campo di testo. Le richieste devono essere gestite direttamente dalla pagina JSP.

Tutte le pagine JSP devono contenere due tasti che permettono (eventualmente tramite Servlet) di passare da **nuovoLibro.jsp** a **elencoLibri.jsp** e viceversa.

Tecnologie Web T
Simulazione Compito

ESERCIZIO 2 (11 punti)

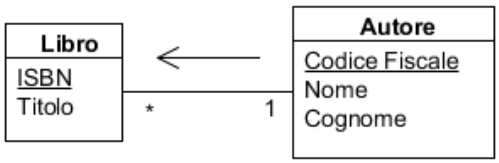
Realizzare una applicazione per la ricerca di case in affitto basandosi principalmente sulla tecnologia Javascript.
A tal fine si realizzi una pagina HTML tramite cui sia possibile inserire la città in cui si vuole fare la ricerca ed il range di prezzo di interesse (minimo e massimo in due campi separati). Durante l'immissione dei campi minimo e massimo, si controlli la validità dei dati inseriti (devono essere valori numerici); in caso di errore si elimini il carattere errato. Inoltre, ogni volta che si inserisce un carattere nel campo **città** si invii tramite AJaX una richiesta asincrona ad una semplice Servlet il cui unico scopo è quello di fornire, dato un parametro **nomecitta** con valore **city**, un documento XML corrispondente al file **city.xml** presente nella directory **resources** (restituire un documento XML col solo tag **<citta>** se non esiste il rispettivo file). In particolare, si consideri il file di esempio **bologna.xml** di seguito:

```
<?xml version="1.0" encoding="UTF-8"?>
<citta nome="bologna">
    <casa>
        <indirizzo>via Sant Isaia 42</indirizzo>
        <prezzo>650</prezzo>
    </casa>
    <casa>
        <indirizzo>piazza San Francesco 45</indirizzo>
        <prezzo>750</prezzo>
    </casa>
    <casa>
    <indirizzo>piazza Malpighi 40</indirizzo>
        <prezzo>720</prezzo>
    </casa>
</citta>
```

Una volta ricevuto il documento XML, lo si salvi in un opportuno oggetto **Mapping** che memorizza documenti XML associati a chiavi alfanumeriche sotto forma di array.
Infine si fornisca un pulsante **show** che, quando premuto, visualizzi indirizzo e costo degli affitti nella città in esame con affitto compreso tra i valori minimo e massimo specificati dall'utente. A tal fine, si utilizzi l'oggetto **Mapping** per recuperare il documento XML corrispondente (quindi senza interagire con la Servlet) e poi, se tale documento esiste, lo si parsifichi opportunamente per ottenere le informazioni indirizzo e prezzo delle case che rispettano i requisiti dell'utente.

ESERCIZIO 3 (11 punti)

Partendo dalla realtà illustrata nel diagramma UML di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su **Hibernate** in grado di "mappare" efficientemente il modello di dominio rappresentato dai Java Bean **Libro** e **Autore** con le corrispondenti tabelle **Libro** e **Autore** contenute nel database **TW_STUD** di **DB2**.



Nel dettaglio, dopo aver creato gli schemi delle tabelle all'interno del proprio schema nel database **TW_STUD**, implementato i Java Bean e definito i file XML di mapping Hibernate, si richiede la realizzazione di una classe di prova facente uso delle API Hibernate in grado di:

- inserire uno o più autori;
- inserire uno o più libri;
- restituire, partendo dal codice fiscale di un autore, i titoli dei libri ad esso associati.

N.B. La relazione 1-N tra Autore e Libro deve essere specificata esplicitamente nel file XML di mapping.